Performance Work Statement

Section 1: Background

For over a decade, Challenge.gov has enabled Federal agencies to foster innovation and competition within the Government by crowdsourcing ideas and solutions from the public. The General Services Administration (GSA) has leveraged Challenge.gov to significantly further their mission to “deliver value and savings in real estate, acquisition, technology, and other mission-support services across government” and promote “effective and efficient government for the American people.” Given the large number of Federal agencies and competitors Challenge.gov is tasked with serving, GSA requires a partner who can support ongoing development of the Challenge.gov platform that will give it more robust knowledge sharing, crowdsourcing, and challenge submission capabilities. As a team of digital services firms with more than two decades of combined experience in agile delivery, DevOps, and application development experience, Team Fearless (comprising Fearless Solutions [Fearless] and SmartLogic) is just such a partner.

1.1 About Team Fearless

Fearless is a HUBZone- and 8(a)-certified full stack development firm based in Baltimore, Maryland, that uses leading-edge agile practices and streamlined DevOps pipelines to rapidly and incrementally deliver transformative solutions to our customers. We pride ourselves on maintaining a team that can work across the entire technology stack, and our experienced team specializes in Design, Development, DevOps and Cloud, Data Analytics, Geospatial, and Agile Coaching. Fearless was started with the goal of streamlining government software, and we’ve never looked back. We believe that everyone benefits when the Government works more efficiently, and we’ve worked with Government customers such as GSA, the Department of Defense, Small Business Administration (SBA), Centers for Medicare and Medicaid Services, Baltimore City Health Department, and the Maryland Department of Commerce.

SmartLogic, a Baltimore-based agile software development firm, has been building web and mobile applications for clients ranging from funded startups to Fortune 500 companies for 14 years. We deliver best-of-breed agile development services and code to our customers by focusing on constant communication, high levels of organization and transparency, thorough planning, and high-quality code developed through best practices such as source code control and continuous integration.

Section 2: Management

Our inclusive management approach is designed to work for everyone, from our QA interns to the end-user, and it is deeply informed by the U.S. Digital Services (USDS) Playbook. Designed to increase the success rate of Government digital services projects by outlining industry best practices, the USDS Playbook provides a path toward the rapid modernization of U.S. Government digital tools and applications. We wholeheartedly embrace each digital play outlined by USDS, and we will use them to guide our approach to the Challenge.gov effort and provide GSA with continuous, streamlined agile support that meets all requirements.

2.1 Agile Processes

Though Team Fearless is well-versed in numerous agile methodologies, we have found it most effective to use a custom Scrum process to offer scalable results that meet customer needs (Figure 1). Our robust communication stack provides for internal and external communications and is supported by thorough requirements management accomplished through flexible software management tools, such as Jira to track sprints (i.e., iterations).

The strength of our Scrum process begins with the team. Teams are built on the skills necessary to complete the work, with the Scrum Master leading agile ceremonies in 2-week sprint cycles. A Scrum Team includes a decision-maker from the customer (i.e., the Product Owner [PO]) who provides feedback and organizes next steps through the product backlog. The initial objective, strategy, and tactics for the project are established through collaborative user story workshops, which are then developed into the product backlog (i.e., to do list). Active customer involvement is one of the key tenets of Scrum, and the PO will be responsible for organizing next steps by prioritizing the backlog during bi-weekly Sprint Planning meetings. Daily standups ensure that the team stays in sync and on track, and we will host a team-level retrospective after each sprint to improve processes and incorporate lessons learned.

Other key agile ceremonies that our team will observe include backlog grooming and sprint reviews. Agile ceremonies enable us to build communication into our schedules, and encourage constant collaboration across the team, regardless of location. Our strong and open communication style allows us to easily manage all team members, whether on-site, off-site, or remote. These processes are supported by our robust DevOps approach, which includes automated testing and continuous deployment best practices, ensuring that new functionality is continuously pushed to production. Our custom Scrum approach creates a tight feedback loop between our teams and stakeholders, ensuring that we continuously deliver powerful infrastructure functionality while providing full accountability to our customers.

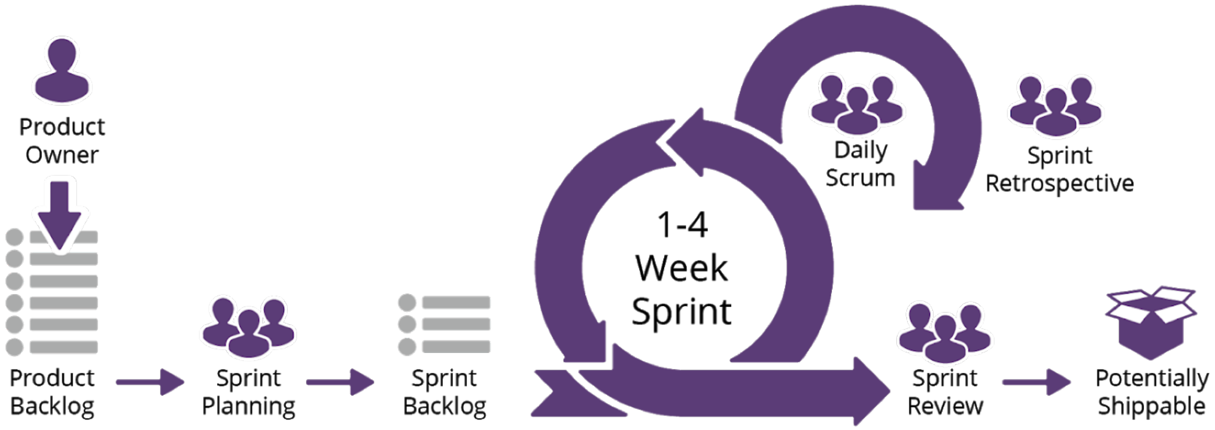

Figure 1: Our iterative agile process enables us to quickly deliver value to the customer, pivot as project needs change, and continuously improve on the product and process.

2.1.1 User Stories

Team Fearless believes that Scrum Teams are responsible for estimating the effort needed to complete items in the product backlog, and they often use story points to assess time requirements and difficulty. This backlog is a living document that grows and evolves over the life of the project. While the responsibility for estimations falls on the team, anyone can add to the product backlog at any time, and only the Product Owner can assign priority. Once priority has been established, the team can evaluate user stories and break them into manageable subtasks during sprint planning or backlog grooming meetings. If a user story is too large, it can be broken down into smaller stories or grouped with related stories in an Epic.

New tasks are typically not added during sprint execution, but a few exceptions apply:

- **All Sprint User Stories Completed** — If all stories in a sprint are completed early, the Scrum Team may pull tickets in from the top of the product backlog or end the sprint early. If the original sprint is ended early, the team completes the sprint review and retrospective and begins a new sprint.
- **Sprint Goal Can No Longer Be Met / Stories Are Roadblocks** — If the Scrum Master realizes that the sprint goal can no longer be met or that user stories are severely road-blocked, there are two courses of action: they can pull in stories from the top of the product backlog or formally end the sprint.

Our team leverages sprint burndown charts to track sprints and assess the amount of work until the project is complete. The burndown chart is updated at the end of each sprint, allowing us to accurately visualize completed tasks and tasks added.

2.1.1.1 User Story Sizing

Estimating story points in order to accurately size each user story is a collaborative and evolving task. Team Fearless has a wealth of experience to leverage when translating user stories into development tickets, and we follow successful agile practices in this regard. First, we get the people downstream of development involved, as they have a deep understanding of past errors that can explode estimates. Next, we work to come to a consensus on the difficulty of each individual task, helping to even out extremes and inviting brief discussion. Utilizing tools like planning poker, our team works in a collaborative manner to estimate stories in the backlog that will be pulled into the upcoming sprint.

2.1.1.2 User Story Prioritization

Once all stories have been estimated, our team commits to what we will complete in the upcoming iteration. This decision is influenced by the following:

- **The priority, as determined by the PO** - Stories with the highest priority are iterated first.
- **Estimation experience** - As the team gains experience with Challenge.gov user stories, their estimations will become increasingly accurate.
- **Prior iterations** - If stories were not completed in a previous iteration, that work needs to be addressed in the current iteration.
- **Dependencies** - We strive to make user stories independent; however, sometimes it is impossible to avoid dependencies, which must be considered when planning an iteration.
- **Story point baseline** - The team should “calibrate” each sprint by using a reference sprint that contains historic story point baselines. This ensures that something they pointed as a “2” previously isn’t pointed as a “5” in a subsequent sprint.

2.1.1.3 Definition of Done

One of the key tenets of agile is to clearly define “Done” as a part of sprint planning from user stories. “Done” should be documented directly on the ticket so that the expected, acceptable end state is clear. This could include performance metrics, a specific deliverable, or how a feature should behave. To promote effective development, Team Fearless will work with the Challenge.gov PO to document and define “Done” for every user story, and the PO has the sign-off authority on the completion of each user story. Below is the list of criteria that Team Fearless validates to ensure stories within a sprint are “Done” and documented:

1. All stories satisfy acceptance criteria
2. All acceptance tests pass (automated where possible)
3. All unit tests pass (automated where possible)
4. All code has been checked in Software Configuration Management
5. All stories accepted as done by the PO
6. All Sprint goals have been achieved

Section 3: DevOps

Team Fearless' design and development approaches are supported by industry-recognized best practices in DevOps, a collaborative process that integrates development and operations through continuous integration, automated testing, and continuous deployment. DevOps processes pair well with Scrum's rapid, communicative approach to development, creating a seamless pathway between our management and technical methodologies. Our DevOps process includes the following:

1. **Software Configuration Management:** Leverage strong version control with tools such as Git (via GitHub).
2. **Continuous Integration:** Deploy and configure a continuous integration server (i.e., Jenkins or Codeship) to help automate the integration of software to discover issues faster in the development lifecycle.
3. **Automated Testing:** Configure automated tests at the unit, functional, and user interface level.
4. **Continuous Deployment:** Configure continuous development to support both continuous delivery and continuous deployment.

3.1 Security

Creating secure software is a key aspect of our software development process. We will create a System Security Plan describing how we will leverage security best practices from among the following throughout both the development lifecycle and delivery to ensure the security of our solutions:

- **Code and Server Access:** Using tools such as PuTTY or KeyGen, we ensure that our codebase, development, and testing servers are accessible by Secure Shell (SSH) keys.
- **Code Quality:** Code reviews / pair programming prevent us from integrating faulty code.
- **Security by Design:** We design security practices into our architecture by coding against incidents like SQL Injection and using Cross Site Scripting as the default implementation.
- **Outdated Servers and Libraries:** Patches and vulnerability upgrades ensure that security measures are up to modern standards.
- **Authorization:** Centralized user authorization and role-based access ensure that users can be granted permissions in one location and ripple through the application.
- **Least Privilege Mentality:** We begin with the idea that restricted access is the default behavior. Escalation of privileges is on an as-needed basis.

We will leverage the aforementioned tools and DevOps and CI / CD best practices to enable automated security testing, continuous monitoring, and other security controls that ensure the product conforms to FedRAMP guidelines. We will also ensure that our development environment follows all guidelines necessary to obtain Authority to Operate (ATO).

3.2 Monitoring / Maintenance

Team Fearless understands that building solutions that will grow with our customers' evolving needs requires the use of a modern tech stack, and we leverage tools like Amazon Web Services (AWS) and Docker to architect reliable, cloud-based solutions that can scale to meet the diverse and changing needs of our customers. We will manage all Challenge.gov cloud infrastructure using popular configuration management tools like Terraform. Our team has deep experience supporting infrastructure best practices for existing systems such as rolling deployments with no downtime, continuous delivery, and high availability.

We use automated monitoring systems like AWS CloudWatch and New Relic to ensure applications are available and performant and create alerts via PagerDuty when performance metrics don't meet our strict service level expectations. We'll use a deployment strategy such as Blue / Green deployment or Canary to limit all unplanned outages. This will allow deployments to production, dev, and staging environments at any time without disrupting users.

Section 4: Design

Once we define the problem space, users, and quality criteria, we use this information to guide our entire development process. Our human-centered design (HCD) methodology ensures that we create products that are desirable and satisfy end-user goals; viable and harmonious with business models and objectives; and buildable, being both technically and operationally feasible. Although HCD activities vary from project to project, our process always flows through the following four phases:

1. **Discovery:** Perform qualitative and quantitative user research (e.g., user interviews, focus groups, card sorting, and A / B testing) to define the problem space from the business, technical, and user experience perspectives. Use this information to refine the project Scope, Goals, and Roadmap.
2. **Modeling:** Define user problems and create user personas, flows, and journey maps. Brainstorm solutions with mind mapping and word association exercises, exploring the full range of possibilities before moving forward with a high-level concept.
3. **Prototyping:** Use wireframes, mockups, and prototypes to test and tweak features and functionality. Iteratively improve the design with increasing levels of detail.
4. **Validation:** Test solution with users, reassessing goals and adjusting as needed. Verify that your solution is compatible with existing design standards and create new visual assets as needed.

4.1 508 Compliance

Team Fearless has extensive experience providing solutions that meet Section 508 Compliance standards, and we are fully prepared to assist GSA in these efforts under the Challenge.gov contract.